



The Perl Debugger

Richard Foley
Richard.Foley@rfi.net

I divide this article into three sections. The first section is a short introduction to the main features of the Perl debugger and how to call it. Next, I describe several basic commands which cover essential debugger usage. Finally, I finish with some of the more interesting commands that have been added to the debugger's repertoire.

Starting the Perl Debugger

The first thing to do is to start my program using the debugger. In my case I'll use the convenient and predictable "Hello World" program with several minor alterations to get a bit more mileage out of it:

```
#!/usr/bin/perl
use strict;

mysub( 'Hello', @ARGV );

sub mysub {
    my @args = @_;
    my @uppd = map { ucfirst($_) }
        @args;
    my $args = join(' ', @uppd);
    print "$args!\n";
}
```

I start the Perl debugger by supplying the `-d` switch to `perl` at run-time:

```
perl -d hello_world.pl World
```

Perl then runs `hello_world.pl` under the Perl debugger with the arguments I provided on the command line. The debugger checks the syntax of the program before running it, loads all libraries required, and stops at the first executable line of the program. In *Listing 1*, the debugger waits for command input.

Notice that the debugger went right past the `use strict` line, because `use` is a *compile-time* directive, and stopped at the first *run-time* statement.

Getting Help

Although familiar with the debugger, I still use the help menu to refresh my memory of precise syntax from time to time. For a help window designed to fit in a typical terminal screen, I use `h`, and for help on a particular command supply the command name as an argument to `h`. So, to get help on the breakpoint command, `b`, I would use `h b`, as in *Listing 2*.

The breakpoint command has a plethora of syntax options. This is one of the things which often puts people off from starting to learn how to use the debugger. It is worth persevering because the commands are mostly quite intuitive. You might also want to get my *Perl Debugger Pocket Reference* and keep it close to your desk.

For the entire help page, I use the `h h` command combined with the `|` (pipe character, for the pager) command, since the output is several normal-sized screens long. The `|` catches the output of the command and presents it to your through your chosen pager so I can see it one screen at a time.

The `|` command is particularly useful when looking at the dump of large data object, for example, as well as for looking at the built-in help pages. I won't show the output because that would use up the available space for this article, but here's how I get the whole help command: `| h h`. An extract of its output in *Listing 2*.

Even the Debugger Uses the Print Command

Many Perl programmers appear to be of the opinion that the `print` command is in some mysterious way better than the debugger. As a serious comment, this is kind of difficult to understand because the debugger is simply another tool to use and not a magic wand. If a tool helps me to get my job done, or done quicker, then I should use it. The debugger of course is not so fickle or so proud that it will not use the available tools. It realizes that the `print`